NPS-OR-06-008

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

Scheduling Policies for an Antiterrorist Surveillance System

by

Kyle Y. Lin
Moshe Kress
Roberto Szechtman

July 2006

Approved for public release; distribution is unlimited.

Prepared for: Naval Postgraduate School
Monterey, CA 93943-5000

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5001

RDML Richard H. Wells, USN
President

Richard Elster
Provost

This report was prepared for Naval Postgraduate School and funded by Naval Postgraduate School Research Initiation Program.

This report was prepared by:

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** July 2006	**3. REPORT TYPE AND DATES COVERED** Technical Report	
4. TITLE AND SUBTITLE: Scheduling Policies for an Antiterrorist Surveillance System			**5. FUNDING NUMBERS** BORYG
6. AUTHOR(S) Kyle Y. Lin, Moshe Kress, and Roberto Szechtman			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER** NPS-OR-06-008
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE** A

13. ABSTRACT ***(maximum 200 words)***

A surveillance system is designated to detect terrorists in a crowded area to prevent a potential attack. Such a system usually does not have the capacity to screen all the people in the area. If the sojourn time distribution of a terrorist is different from that of the other people in the crowd, then it is possible to increase the probability of detecting a terrorist in time by using a decision rule to choose whom to inspect next. We use a queueing model with impatient customers to describe the interaction between the surveillance system and the surveyed people. We identify a few cases when a simple service rule—such as the first-come-first-serve rule—is optimal. In general, we develop a heuristic policy that is particularly effective in an area with heavy traffic, such as an airport check-in lobby.

14. SUBJECT TERMS Homeland security, counterterrorism, search and surveillance, M/G/1 queue, impatient customers.			**15. NUMBER OF PAGES** 31
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A surveillance system is designated to detect terrorists in a crowded area to prevent a potential attack. Such a system usually does not have the capacity to screen all the people in the area. If the sojourn time distribution of a terrorist is different from that of the other people in the crowd, then it is possible to increase the probability of detecting a terrorist in time by using a decision rule to choose whom to inspect next. We use a queueing model with impatient customers to describe the interaction between the surveillance system and the surveyed people. We identify a few cases when a simple service rule—such as the first-come-first-serve rule—is optimal. In general, we develop a heuristic policy that is particularly effective in an area with heavy traffic, such as an airport check-in lobby.

THIS PAGE INTENTIONALLY LEFT BLANK

1 Introduction

Terrorist attacks—such as bombing, assassination of political figures, and release of poison gas in a crowd—are serious threats in many regions of the world. A significant terrorist attack occured in 1972 at a ticket counter in Lod International airport near Tel Aviv, Israel, where a three-man hit squad from the Japanese Red Army killed 26 people and injured 78 more [1]. More recent examples include the 9/11 attack in 2001, the Bali bombings in 2002 and 2005, and the London bombings in 2005. Numerous instances in the past suggest that terrorists typically aim their attacks at crowded locations—such as restaurants, transportation terminals, popular tourist spots, political rallies, and subway stations—to create chaos and cause damage. The consequences of such terrorist attacks are casualties, damaged property, and a major disruption of daily life.

Response actions for countering terrorist attacks range from politically and socially driven attempts to deter recruitment of terrorists, through intelligence-based arrests of suspects, to surveillance and protection of potential targets. While arguably it would be most effective to go after the sources of such attacks—the terrorist organizations and their infrastructure—implementing such a policy has not been very effective so far. Consequently, the authorities still need to protect the public from the end-operators of such organizations (assassins, suicide bombers, and the like) by timely detection and effective response. In this paper we focus on that last line of defense—the problem of detecting, as early as possible, a terrorist attack on a public target.

Several attempts have recently been made to model and analyze detection and response actions associated with counterterrorism and homeland security. Jacobson et al. [6] consider the problem of baggage screening procedures at airports. Their objective is to optimize the performance of this process subject to a finite amount of screening resource. McLay et al. [11] model a multilevel screening process for airline passengers that are subject to profiling. Wein and Liu [17] analyze a single facility milk production/distribution supply chain that is the victim of a deliberate release of botulinum toxin. They conclude that a timely and specific in-process detection policy has the potential to eliminate the threat of this scenario at a relatively low cost. Kaplan et al. [8] combine a stochastic model of blood donation with an epidemic model to assess whether screening blood donors could provide an early warning of a bioterror attack. Kress [9] develops a model that estimates the effect of a suicide attack, based on which Kaplan and Kress [7] model and analyze a detection scheme for suicide bombers in urban settings.

1.1 Research Problem

We consider a large, enclosed area—henceforth called *arena*—such as an airport check-in area or a popular tourist attraction. An array of video cameras monitor the arena and feed real-time video streams to a control center, where a security team screens people in two phases. In the first phase, the people in the arena are examined with naked eyes and each person is immediately put into one of two groups: *nonsuspects* and *suspects*. Only suspects are subject to the second-phase screening, which includes taking their biometric signature (such as face structure, hair color, etc.) and running it through a terrorist database for

comparison. In case of a positive match, the suspect is classified as a *potential terrorist* and security forces are notified to take proper actions; otherwise, the suspect is reclassified as a nonsuspect and the security team moves on to conduct the second-phase screening on another suspect. The problem is to determine which suspect the security team should go after first, when faced with many suspects who are subject to the second-phase screening.

Two observations motivate our research problem. First, because the second-phase screening takes time, the security team may not be able to inspect all suspects before they leave the arena. Second, because a terrorist's intention and action are different from those of other people in the arena, his sojourn time distribution may be different too. Consequently, by carefully choosing which suspect to inspect next, one could potentially increase the probability of detecting a terrorist in time. In this paper, we develop a queueing model with impatient customers to analyze this problem, and draw insights into the effect of scheduling policies on such a surveillance system.

1.2 Overview and Outline

The contribution of this paper is twofold. From a theoretical standpoint, we build a queueing model with impatient customers that describes the antiterrorist surveillance system. There are two types of customers—terrorists and nonterrorists. The novelty of this queueing model is that only one type of customer (terrorists) is worth serving, but the server does not know a customer's identify until service completion. From an application standpoint point, we develop scheduling policies for an antiterrorist surveillance system that can improve the probability of detecting a terrorist in a crowded area.

The rest of this paper is organized as follows. In Section 2 we discuss the operational setting and develop a queueing model that describes it. In Section 3 we discuss the case when the sojourn times of customers follow exponential distributions, and identify a few cases where the optimal policy can be explicitly determined. In Section 4 we consider general sojourn time distributions, and develop a heuristic policy that is particularly effective in heavy-traffic situations. Conclusions and future research directions are discussed in Section 5.

2 The Operational Setting and a Queueing Model

In this section, we describe the operational setting and formulate the second-phase screening problem as a single-server queueing control problem.

Consider a terrorist who attempts an attack in a crowded area, called arena, such as an airport check-in hall. The objective of the surveillance system is to detect the terrorist in time so that the security forces can take proper actions to prevent or mitigate the attack. People who arrive to the arena are immediately classified as *suspects* or *nonsuspects*. The suspects, who arrive according to a Poisson process, generate a queue with respect to the second-phase screening of the surveillance system. Each suspect is independently a red customer (terrorist) with probability p, or a white customer (nonterrorist) with probability $1 - p$. A customer (suspect) leaves the queue after a random amount of time independent of whether service (second-phase screening) has started. The sojourn time of a white customer, which

is the time between arrival to the arena and its departure, follows a distribution function $F_W(\cdot)$. The sojourn time of a red customer, which is the time between arrival to the arena and the moment he initiates the attack, follows a distribution function $F_R(\cdot)$.

The *service* at the second-phase comprises a continuous monitoring of the suspect, while running the suspect's biometric signature through a terrorist database for comparison. The security team is the *server*, which can serve one customer at a time. The service times (for database search) are independent and identically distributed random variables with distribution function $F_S(\cdot)$. We assume that if the surveillance system completes the screening of a red customer before he initiates the attack, then the attack is prevented. If the red customer initiates the attack while in service, the continuous monitoring enables the system to detect it and instigate a quick response (e.g., instructing the crowd to "hit the deck") such that the attack is mitigated.

White customers represent nonterrorists in the arena. If a white customer leaves the arena before ever entering service, he simply goes away and the queue length is reduced by one. If a white customer leaves the arena while in service, the queue length is reduced by one and the server becomes available. If a white customer completes the service, the server records his data and removes him from the surveillance queue. Because of the recorded data, this white customer will not be served again.

Although the goal of the server is to serve a red customer, the server does not know the customer's identity until the service is completed or when a red customer initiates an attack during service. If a red customer initiates his attack before entering service, then the surveillance system *fails* because it cannot prevent or even mitigate the attack. If a red customer enters service before initiating the attack, then the attack will be prevented (if he completes service) or mitigated (if he initiates the attack while in service). In either case, we assume that the surveillance system *succeeds*. The problem ends when a red customer departs the queue for the first time—either due to service completion or due to attack initiation. The *objective function* is to find a service discipline that *maximizes the probability that the first departing red customer has entered service before initiating the attack.*

The server's problem is to decide which customer in the queue to serve each time the server becomes available. Specifically, the state of the queue is given by

$$(t_1, t_2, \ldots, t_n), \qquad t_1 > t_2 > \cdots > t_n,$$

with the interpretation that there are n customers in the queue, and the ith customer has stayed in the queue for t_i time units. Note that we do not need to include the time since the last customer arrival in the state space because the customer arrival process is a Poisson process. A feasible policy is a function that maps a vector $(t_1, \ldots, t_n)$ to an index $i \in \{1, \ldots, n\}$, for $n = 1, 2, \ldots$. Although the model allows p to be an arbitrary number between 0 and 1, we are most interested in the case when p is close to 0.

In queueing theory, there is extensive research that concerns dynamic scheduling of a multiclass queue. In a service center, different classes of customers bring in different revenue and require different service times; for example, see Miller [12] and Harrison [4]. In a production system, switching from one customer class to the other may require setup times; for example, see Reiman and Wein [15] and Olsen [13]. More recently, there is a growing

interest in a multiclass queue in heavy traffic; for example, see Bertsimas and Mourtizinou [2], Plambeck et al. [14], and Harrison and Zeevi [5]. The major distinction of our model from these earlier works is that a customer does not reveal his identity upon arrival, while the server can gather information about a customer's identity by studying his sojourn time. To the best of our knowledge, our work is the first to address this type of problem.

3 Exponential Sojourn Time Distribution

This section presents the case when both F_R and F_W are exponential. In Subsection 3.1 we study the *first-come-first-serve* rule, and in Subsection 3.2 we study the *last-come-first-serve* rule. In Subsection 3.3 we consider the *random-selection* rule, and compare all three rules numerically. Although our primary interest is to study a nonpreemptive service system, in Subsection 3.4 we discuss a preemptive service system that complements our theoretical results.

3.1 First-Come-First-Serve (FCFS) Rule

According to the FCFS rule, the server always serves the customer who has stayed the longest in the queue. If the sojourn time distributions for both red and white customers are exponential, the next theorem presents a sufficient condition for the FCFS rule to be optimal. Note that the theorem does not require the service time distribution F_S to be exponential, neither does it require the arrival process to be a Poisson process.

Theorem 3.1 *If both F_R and F_W are exponential with respective rates $\theta_R < \theta_W$, then the FCFS rule is optimal for an arbitrary distribution function F_S and for an arbitrary arrival process.*

Proof: Consider an arbitrary state $(t_1, t_2, \ldots, t_n)$ such that $t_1 > t_2 > \cdots > t_n$. To prove the theorem, we will show that for each policy that does not serve customer 1 first, we can find a better policy that does start with customer 1. The proof relies on an argument that involves stochastic coupling between two sample paths. A reference to the stochastic coupling technique can be found in Section 9.2 in Ross [16].

Consider two servers—server A and server B—each facing the state $(t_1, \ldots, t_n)$. Suppose server B uses a policy ϕ, in which $\phi(t_1, \ldots, t_n) = i \neq 1$. Consider a policy for server A as follows: Serve customer 1 first. If server A finds customer 1 to be white and no red customer has left (unserved) yet, then (1) if customer i is not in the queue, switch to policy ϕ thereafter; (2) if customer i is still in the queue, then relabel customer i as customer 1 and switch to policy ϕ thereafter.

Let $p(t)$ denote the probability that a customer in the queue is red if he has stayed in the queue for t time units. Using Bayes' rule, we can calculate that

$$p(t) = \frac{p\bar{F}_R(t)}{p\bar{F}_R(t) + (1-p)\bar{F}_W(t)}, \tag{1}$$

where $\bar{F}_R(t) \equiv 1 - F_R(t)$ and $\bar{F}_W(t) \equiv 1 - F_W(t)$ are the tail distribution functions of a red and white customers' sojourn time, respectively. Because both F_R and F_W are exponential with respective rates $\theta_R < \theta_W$, it follows that $p(t)$ increases in t (the first derivative of Equation (1) is positive). Therefore, we have that $p(t_1) > p(t_2) > \cdots > p(t_n)$.

Let servers A and B serve queues A and B, each in state $(t_1, \ldots, t_n)$, respectively. Because $p(t_1) > p(t_i)$, we are able to couple customer 1's identity and customer i's identity in queues A and B in the following 5 cases:

1. With probability $p(t_1)p(t_i)$, customers 1 and i in both queues are red: Because of the memoryless property of exponential distributions, the remaining sojourn times of customers 1 and i are identically distributed in both queues. Therefore, both servers will eventually catch the red customer in service if no other red customer leaves (unserved) sooner. As a consequence, the probability that the first red customer leaving queue A will receive (at least partial) service is the same as that in queue B.

2. With probability $(1 - p(t_1))(1 - p(t_i))$, customers 1 and i in both queues are white: In this case, we couple customer 1's (i's) remaining sojourn time in queue A and customer i's (1's) remaining sojourn time in queue B. In addition, we couple the identity of all other customers and their respective remaining sojourn times in two queues, as well as the arrival process and sojourn times of future customers. Finally, we couple the service times for the two servers such that the kth service initiated by server A takes the same amount of time as the kth service initiated by server B. By doing so, we can see that both queues will follow the same sample path—except that the labels of customers 1 and i are swapped. Consequently, the probability that the first red customer leaving queue A will receive (at least partial) service is the same as that in queue B.

3. With probability $(1 - p(t_1))p(t_i)$, customer 1 in queue A and customer i in queue B are red, and customer i in queue A and customer 1 in queue B are white: Similar to case 1, the two servers will do equally well in terms of the probability of success.

4. With probability $(1 - p(t_1))p(t_i)$, customer 1 in queue A and customer i in queue B are white, and customer i in queue A and customer 1 in queue B are red: Similar to case 2, two servers will do equally well in terms of the probability of success.

5. With probability $p(t_1) - p(t_i)$, customer 1 is red in both queues A and B, and customer i is white in both queues A and B: In this case, we couple the customer identities for the other $n - 2$ customers in queues A and B, and all their future arrivals. Because server A starts with a red customer and server B starts with a white customer, it follows that the probability that the first red customer leaving queue A will receive (at least partial) service is greater than that in queue B.

In summary, in cases 1–4, the two servers do equally well, while in case 5 server A can do better than server B. It follows that there exists a policy that can do better than policy ϕ by immediately serving customer 1—the customer who has stayed in the queue for the longest period of time. □

Although Theorem 3.1 holds for an arbitrary value of p, we are particularly interested in the case when $p \to 0$, because a terrorist attack is a rare event. To compute the objective function as $p \to 0$, we first construct a queue with only white customers arriving according to a Poisson process with rate λ, and then let a red customer arrive in steady state. If the service time distribution also follows an exponential distribution, then we can obtain a closed-form solution for our objective function—the probability that the red customer who arrives in steady state will enter service before leaving the queue (initiating the attack).

Let μ denote the rate of the exponential distribution associated with the service times. With white customers arriving according to a Poisson process with rate λ, the steady-state probability that there are n customers in the queue can be found by a Markov chain argument (see Donald and Harris [3] for a derivation), and is given by

$$\frac{1}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})}, \qquad \text{for } n=0, \tag{2}$$

and

$$\frac{\prod_{i=1}^{n}(\frac{\lambda}{\mu+i\theta_W})}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})}, \qquad \text{for } n=1,2,\ldots.$$

If a red customer finds n white customers in the queue upon arrival, then the probability that he will enter service before leaving is the probability that all those n white customers depart—either due to impatience or due to service completion—before the red customer does. This probability can be obtained by the memoryless property of the exponential distributions:

$$\prod_{i=1}^{n}\Big(\frac{\mu+i\theta_W}{\mu+i\theta_W+\theta_R}\Big).$$

Therefore, with the FCFS rule, the probability that the red customer arriving in steady state will enter service before leaving is

$$\frac{1}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})}+\sum_{n=1}^{\infty}\Big[\Big(\frac{\prod_{i=1}^{n}(\frac{\lambda}{\mu+i\theta_W})}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})}\Big)\prod_{i=1}^{n}\Big(\frac{\mu+i\theta_W}{\mu+i\theta_W+\theta_R}\Big)\Big].$$

We plot the preceding in Figures 1 and 2 with different parameters, which will be discussed later on at the end of Section 3.3.

3.2 Last-Come-First-Serve (LCFS) Rule

With the LCFS rule, the server always serves the customer who most recently joined the queue. Somewhat surprisingly, the counterpart of Theorem 3.1 when $\theta_R > \theta_W$ is not true even if F_S is also exponential. For example, if there is only one customer in the queue, and that customer has been in the queue for a long time so that it is most likely white, then the server may prefer waiting for the next new arrival rather than serving that very old customer, as shown in the next example.

Example 3.1
Suppose $\lambda = \mu = 1$, $\theta_R = 10$, $\theta_W = 0.1$, and $p = 0.8$. Consider a situation when there is only one customer in the queue—referred to as customer Z throughout this example—who joined the queue one time unit ago. Recall from Equation (1) that customer Z is a red customer with probability $p(1) \approx 0.0002$.

With the LCFS rule, the server initiates service with customer Z. Let A denote the event that the first departing red customer will receive service under the LCFS rule. We can compute $P\{A^c\}$ by conditioning on the identity of customer Z:

$$\begin{aligned} P\{A^c\} &= p(1)P\{A^c|\text{customer Z is red}\} + (1-p(1))P\{A^c|\text{customer Z is white}\} \\ &> (1-p(1))\frac{\lambda}{\lambda+\theta_W+\mu}\, p\, \frac{\theta_R}{\theta_R+\theta_W+\mu} \\ &\approx 0.34, \end{aligned}$$

where the inequality follows because conditional on customer Z being white, event A^c occurs as long as the following three events occur sequentially: (1) a new customer arrives before customer Z departs (whether due to impatience or due to service completion); (2) the new customer is red; and (3) the new customer leaves (unserved) before customer Z departs. Hence, we have that $P\{A\} < 0.66$.

An alternative policy is for the server to stay idle until a new customer arrives, and then immediately serve the newly arrived customer. Let B denote the event that the first departing red customer will receive service under this policy. We can compute $P(B)$ by conditioning on the identity of customer Z:

$$\begin{aligned} P\{B\} &= p(1)P\{B|\text{customer Z is red}\} + (1-p(1))P\{B|\text{customer Z is white}\} \\ &> (1-p(1))\, p\, \frac{\theta_R+\mu}{\lambda+\theta_R+\mu} \\ &\approx 0.73, \end{aligned}$$

where the inequality follows because conditional on customer Z being white, event B occurs as long as the first arrival is a red customer, and that red customer departs (either due to impatience or due to service completion) before another new customer arrives.

Finally, because $P\{B\} > 0.73 > 0.66 > P\{A\}$, it follows that the LCFS rule is not optimal. □

Similar to the FCFS rule discussed in Section 3.1, we next let $p \to 0$ and calculate the objective function—the probability that the first red customer departing the queue receives (at least partial) service. As $p \to 0$, we can find this probability by first constructing a queue with only white customers and letting a red customer arrive in steady state. The objective function in the case of $p \to 0$ becomes the probability that a red customer arriving in steady state will enter service before leaving the queue.

First note that if the server is idle when a red customer arrives, then the red customer enters service immediately. If the server is busy when a red customer arrives, then with the LCFS rule, the current number of white customers in the queue is irrelevant to whether the

red customer will enter service before leaving. In order to obtain the probability that the red customer will enter service before leaving if the server is busy upon the red customer's arrival, we next formulate a Markov chain.

Suppose the server is busy when a red customer arrives and joins at the end of the queue. Define a Markov chain to represent the state of the queue when a red customer is present. Denote by k the state if the server is busy with a white customer, and there are $k-1$ white customers in the queue *behind* the red customer, $k = 1, 2, \ldots$. Let the state become 0 when the red customer enters service, and -1 when the red customer leaves before entering service. Note that by definition, states 0 and -1 are absorbing, and that the Markov chain starts in state 1.

Let α_k, $k = 1, \ldots, \infty$, denote the probability that the Markov chain in state k will *ever* enter state $k-1$ before the red customer leaves (entering state -1). By definition, α_1 is the probability that a red customer will enter service before leaving if the server is busy upon his arrival.

To obtain α_1, we first find an expression for α_k by conditioning on whether the next event is a new arrival, a departure of a white customer, or the departure of the unserved red customer (initiating an attack):

$$\alpha_k = \frac{\lambda}{\theta_R + \lambda + \mu + k\theta_W} \cdot \alpha_{k+1}\alpha_k + \frac{\mu + k\theta_W}{\theta_R + \lambda + \mu + k\theta_W} \cdot 1 + \frac{\theta_R}{\theta_R + \lambda + \mu + k\theta_W} \cdot 0,$$

for $k = 1, 2, \ldots$. Solving for α_k yields

$$\alpha_k = \frac{\mu + k\theta_W}{\theta_R + \lambda + \mu + k\theta_W - \lambda\alpha_{k+1}}. \tag{3}$$

Because $\alpha_{k+1} \in [0, 1]$, the preceding implies that

$$\frac{\mu + k\theta_W}{\theta_R + \lambda + \mu + k\theta_W} < \alpha_k < \frac{\mu + k\theta_W}{\theta_R + \mu + k\theta_W}. \tag{4}$$

Consequently, we can choose a large value of k, use Equation (4) to bound α_k, and then use Equation (3) to recursively compute the bounds for $\alpha_{k-1}, \alpha_{k-2}, \ldots, \alpha_1$. Because the bounds converge very quickly, we can approximate α_1 satisfactorily.

Finally, we can compute the probability that the red customer enters service before leaving under the LCFS rule by

$$\begin{aligned} &1 \cdot P\{\text{server idle in steady state}\} + \alpha_1 \cdot P\{\text{server busy in steady state}\} \\ &= \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} (\frac{\lambda}{\mu + i\theta_W})} + \alpha_1 \Big(1 - \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} (\frac{\lambda}{\mu + i\theta_W})}\Big), \end{aligned}$$

where the steady-state probability follows from Equation (2). We plot the preceding results in Figures 1 and 2, which will be discussed at the end of Section 3.3.

3.3 Random Selection (RS) Rule

Another service rule of interest is the RS rule, in which the server, when becoming available, selects the next customer at random. If both sojourn times are exponentially distributed and $\theta_R = \theta_W$, then all three rules—FCFS, LCFS, and RS—perform equally well for two reasons: (1) each customer in the queue still has a probability p of being red regardless of the amount of time he has stayed in the queue; and (2) the remaining sojourn times for all customers in the queue are independent and identically distributed because of the memoryless property of the exponential distribution. If $\theta_R \neq \theta_W$, we would expect that the performance of the RS rule lies between those of the FCFS and the LCFS rules.

As $p \to 0$, we can formulate a Markov chain to compute the performance of the RS rule in a similar way to that in Sections 3.1 and 3.2. We omit the derivation, but plot the performance of the RS rule in Figures 1 and 2 for comparison. As seen in these two figures, the FCFS rule is the best of the three when $\theta_R < \theta_W$ (in this case, the FCFS is optimal by Theorem 3.1), while the LCFS rule is the best when $\theta_R < \theta_W$. However, FCFS is the most sensitive to the changes either in θ_W or in θ_R, while for LCFS the slope is relatively mild, especially when $\theta_R \approx \theta_W$. This observation suggests that if the value of θ_R is highly uncertain—as θ_W is typically much easier to estimate—then the LCFS rule is more robust.

3.4 Preemptive Service

Our queueing model assumes that the service is nonpreemptive because the screening process of a suspect cannot be interrupted. Although preemptive service—whereby the server can interrupt its current screening and switch to another customer upon a new arrival or any departure—is not a practical assumption, we are interested in such a variation from a theoretical standpoint. This subsection presents a theorem that complements Theorem 3.1 in the nonpreemptive service case.

Theorem 3.2 *If the service is preemptive, and both F_R and F_W are exponential with respective rates $\theta_R < \theta_W$ (respectively, $\theta_R > \theta_W$), then the FCFS (respectively, LCFS) rule is optimal for an arbitrary arrival process if F_S is exponential.*

Proof: We prove the optimality of the LCFS rule when $\theta_R > \theta_W$, while the proof for the optimality of the FCFS rule when $\theta_R < \theta_W$ follows a similar argument.

Consider two servers—server A and server B—each facing the state $(t_1, t_2, \ldots, t_n)$, such that $t_1 > t_2 > \cdots > t_n$. Suppose server B uses a policy ϕ, in which $\phi(t_1, \ldots, t_n) = i \neq n$. Consider a feasible policy for server A as follows: Start service with customer n and continue until an arrival or departure, and thereafter switch to policy ϕ if the next event is not a departure of a red customer (in which case the problem ends). Let E_A (or E_B, respectively) denote the event that in queue A (or queue B, respectively) the first departing red customer receives (at least partial) service. To prove the theorem, we will show that $P(E_A) > P(E_B)$.

Recall from Equation (1) that $p(t)$ denotes the probability that a customer is red if he has stayed in the queue for t time units. Because $\theta_R > \theta_W$ and $t_1 > t_2 > \cdots > t_n$, it follows that $p(t_1) < p(t_2) < \cdots < p(t_n)$. We couple customer n's identity and customer i's identity

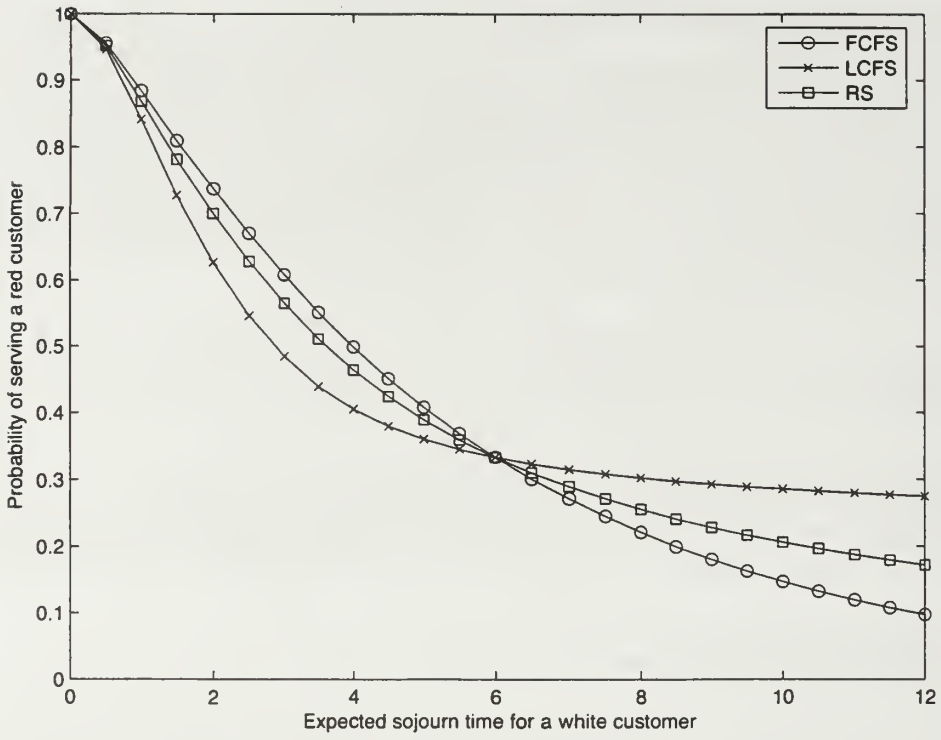

Figure 1: Probability that the first departing red customer receives (at least partial) service as a function of a white customer's expected sojourn time. F_R, F_W, and F_S are all exponential; $p \to 0$, $\lambda = 2$, $1/\theta_R = 6$, and $1/\mu = 2$.

in queues A and B in the following 4 cases, and let the random variable I indicate which case takes place:

1. With probability $p(t_n)p(t_i)$, both customers n and i in both queues are red: Because (additional) sojourn times of red customers follow independent exponential distribution with rate θ_R, it follows that $P(E_A|I = 1) = P(E_B|I = 1)$.

2. With probability $(1 - p(t_n))(1 - p(t_i))$, both customers n and i in both queues are white: Similar to case 1, because (additional) sojourn times for white customers follow independent exponential distribution with rate θ_W, it follows that $P(E_A|I = 2) = P(E_B|I = 2)$.

3. With probability $(1 - p(t_n))p(t_i)$, customer n is white and customer i is red in both queues: We couple the next event that takes place. If the next event is service completion or customer i's (red) departure, then E_B occurs, but not E_A; otherwise, the two

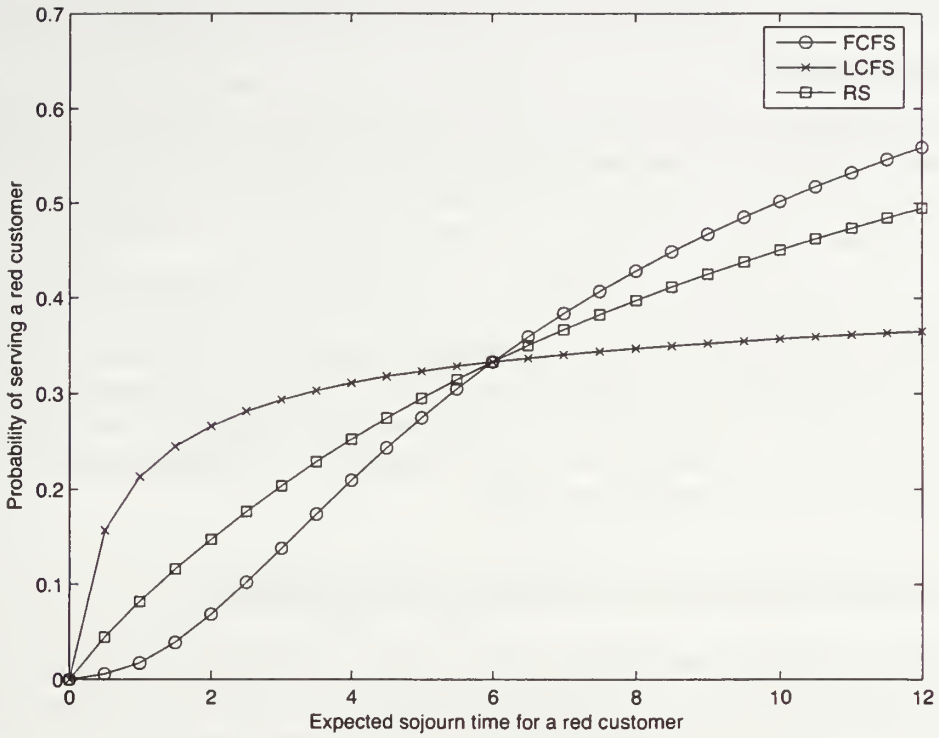

Figure 2: Probability that the first departing red customer receives (at least partial) service as a function of a red customer's expected sojourn time. F_R, F_W, and F_S are all exponential; $p \to 0$, $\lambda = 2$, $1/\theta_W = 6$, and $1/\mu = 2$.

servers will do equally well because the service distribution is exponential. Therefore,

$$P(E_B|I = 3) - P(E_A|I = 3) > 0.$$

4. With probability $(1 - p(t_i))p(t_n)$, customer n is red and customer i is white in both queues: We couple the next event that takes place. If the next event is service completion or customer n's (red) departure, then E_A occurs, but not E_B; otherwise, the two servers will do equally well because the service distribution is exponential. As the situation is reverse to that in case 3, it follows that

$$P(E_A|I = 4) - P(E_B|I = 4) = P(E_B|I = 3) - P(E_A|I = 3) > 0.$$

Finally, we have that

$$P(E_A) - P(E_B) = \sum_{k=1}^{4} [P(E_A|I = k) - P(E_B|I = k)]P(I = k)$$

$$= [P(E_A|I=4) - P(E_B|I=4)][(1-p(t_i))p(t_n) - (1-p(t_n))p(t_i))]$$
$$= [P(E_A|I=4) - P(E_B|I=4)](p(t_n) - p(t_i)) > 0,$$

and the result follows. □

Note that contrary to Theorem 3.1, Theorem 3.2 does not hold for an arbitrary service distribution function F_S. A counterexample can be constructed intuitively as follows. Suppose that F_R and F_W are exponential with respective rates $\theta_R < \theta_W$, and that F_S has a decreasing failure rate—that is, $f_S(t)/\bar{F}_S(t)$ decreases in t, for $t > 0$—so that the longer a customer has been in service, the longer (stochastically) his remaining service time becomes (for example, see Proposition 9.1.3 in Ross [16]). Granted, with the FCFS rule, the server always serves the customer who has the highest probability of being red. However, after serving the same customer for a long time without a conclusion, the remaining service time tends to be even longer (in the regular stochastic sense). At that point, the server may prefer to switch to another customer for a fresh service time, even though the probability for this other customer to be red is slightly smaller.

4 General Sojourn Time Distributions

This section presents the case when F_R and F_W do not necessarily follow exponential distributions. In Subsection 4.1, we discuss the difficulty of finding the optimal policy in the case of general sojourn time distributions, and present an example to demonstrate it. In Subsection 4.2, we develop a heuristic policy, which works particularly well under heavy traffic—the case we expect in real-world applications. In Subsection 4.3, we use simulation to numerically evaluate different policies.

4.1 Difficulty of Finding the Optimal Policy

We first investigate whether we can relax the conditions specified in Theorem 3.1 so that the FCFS rule is still optimal under weaker conditions. Intuitively, for the FCFS rule to be optimal, two conditions must be satisfied: (1) the longer a customer has stayed in the queue, the more likely he is a red customer; and (2) the longer a customer has stayed in the queue, the sooner he tends to leave the queue according to his sojourn time distribution.

For condition (1) to hold, we need $p(t)$ in Equation (1) to increase monotonically in t. Using calculus, we can show that for $p(t)$ to satisfy this property, a sufficient condition is that F_R has a smaller failure rate than F_W; in other words,

$$\frac{f_R(t)}{\bar{F}_R(t)} \leq \frac{f_W(t)}{\bar{F}_W(t)}, \qquad \text{for all } t > 0, \tag{5}$$

where f and g are the density functions and $\bar{F}_R$ and $\bar{F}_W$ are the tail distribution functions.

For condition (2) to hold, we need $(R-t \mid R > t)$ to decrease in t in the regular stochastic sense, where R denotes a random variable with distribution function F_R. That is, we need

$$P\{R - t > s | R > t\} = \frac{\bar{F}_R(t+s)}{\bar{F}_R(t)}$$

to decrease in t for all $s > 0$. Straightforward calculation shows that the preceding decreases in t if and only if $f_R(t)/\bar{F}_R(t)$—the failure rate of F_R—increases in t. Consequently, condition (2) holds if the failure rate function is increasing for both F_R and F_W.

If the failure rate function is increasing for both F_R and F_W, and the failure rate of F_R is smaller than that of F_W, then the longer a customer has stayed in the queue, not only is the customer more likely to be red, but the customer also tends to leave the queue sooner. Hence, it makes intuitive sense for the FCFS rule to be optimal. However, the next example shows that this conjecture is not true in general.

Example 4.1
Suppose that F_R and F_W are identical with the following (increasing) failure rate function:

$$\frac{f_R(t)}{\bar{F}_R(t)} = \frac{f_W(t)}{\bar{F}_W(t)} = \begin{cases} 0, & \text{for } 0 \le t < 2, \\ 1, & \text{for } 2 \le t < 4, \\ 10000, & \text{for } t \ge 4. \end{cases}$$

Also suppose that the probability of a red customer is $p = 0.5$. Because F_R and F_W are identical, the server cannot learn about a customer's identity from the amount of time the customer has stayed in the queue. Hence, $p(t) = p = 0.5$ for all $t > 0$.

Suppose there are three customers with $t_1 = 2.99$, $t_2 = 2.01$, and $t_3 = 1$, and that the service time distribution F_S is deterministic and equal to 1. In addition, assume $\lambda = 0.0001$ so that the effect of future arrivals is negligible. Conditional on the event that at least one of the three customers currently in the queue is red, we use Monte Carlo simulation to compare the service orders 1, 2, 3, and 2, 1, 3. It turns out that the probability the first departing red customer will receive (at least partial) service is 0.720 for the service order 1, 2, 3, and 0.752 for the service order 2, 1, 3 (the standard error of each estimator is less than 5×10^{-5}). Therefore, it is better to start with customer 2 rather than with customer 1, and the FCFS rule is not optimal.

To gain some intuition about this example, first note that the failure rate function remains a constant for $2 \le t < 4$, and the service time is deterministic and equal to 1. Because $[t_i, t_i + 1] \subset [2,4)$ for $i = 1,2$, the time it takes for the server to become available is identically distributed regardless of whether the server starts with customer 1 or customer 2. In addition, if the server starts with customer 1, the probability that customer 2 is still in the queue when the server becomes available is the same as the probability that customer 1 is still in the queue if the server starts with customer 2. Consequently, the number of customers between customers 1 and 2 that the server can serve by following the order 2, 1, 3 is identically distributed to that number when the server follows the order 1, 2, 3.

However, by starting with customer 2, the time it takes for the server to become available for customer 3 is stochastically smaller than by starting with customer 1, because as soon as a customer spends 4 time units in the queue, it will leave almost immediately. Consequently, by starting with customer 2, the server has a better chance to serve customer 3, and therefore with the service order 2, 1, 3, the server has a better chance to serve the first departing red customer. □

4.2 Heuristic Policy

To develop a heuristic policy, recall that our original problem can be stated as follows:

A: Customers arrive according to a Poisson process with rate λ. Each customer is independently a red customer with probability p or a white customer with probability $1-p$. Choose a service rule to maximize the probability that the first departing red customer receives (at least partial) service.

Recall that we are primarily interested in the case when $p \to 0$ because, in reality, a terrorist attack—represented by a red customer—is a rare event. As $p \to 0$, our original problem in A can be restated as follows:

B: Suppose white customers arrive according to a Poisson process with rate λ, and a red customer arrives at the queue in its steady state. Choose a service rule to maximize the probability that this red customer will enter service before leaving the queue.

Consider a related problem:

C: Customers arrive according to a Poisson process with rate λ, where each customer is red with probability p or white with probability $1-p$. Choose a service rule to maximize the long-run proportion of red customers who enter service before leaving.

As $p \to 0$ in problem C, the interarrival times between successive red customers follow an exponential distribution whose mean converges to infinity. Therefore, the number of white customers in the queue when a red customer arrives converges to the steady-state distribution of a system where white customers arrive according to a Poisson process with rate λ. In other words, as $p \to 0$, problem C reduces to problem B. It then follows that problems A and C become equivalent as $p \to 0$. Next, we motivate our heuristic policy by considering problem C.

In problem C, suppose we earn a reward of 1 for each red customer served, so that the objective function becomes maximizing the long-run reward rate. We propose a heuristic policy that chooses the customer with the highest reward rate—the ratio between the expected reward and the expected time spent if the customer is served. Let R, W, and S denote random variables with respective probability distribution functions F_R, F_W, and F_S. Suppose a customer has stayed in the queue for t time units, then serving that customer yields a reward rate equal to

$$\begin{aligned} r(t) &\equiv \frac{P\{\text{customer is red} \mid \text{sojourn time is at least } t\}}{E[\text{additional time in queue if served}]} \\ &= \frac{p(t)}{p(t)E[\min(R-t,S)|R>t] + (1-p(t))E[\min(W-t,S)|W>t]}, \end{aligned}$$

where $p(t)$ is given by Equation (1). As $p \to 0$, we can compare the reward rate between any two customers by

$$\lim_{p\to 0} \frac{r(t_1)}{r(t_2)} = \left(\frac{\bar{F}_R(t_1)/\bar{F}_W(t_1)}{E[\min(W-t_1,S)|W>t_1]}\right) \Big/ \left(\frac{\bar{F}_R(t_2)/\bar{F}_W(t_2)}{E[\min(W-t_2,S)|W>t_2]}\right). \tag{6}$$

Therefore, we define a score function for a t-time-unit-old customer as

$$s(t) \equiv \frac{\bar{F}_R(t)/\bar{F}_W(t)}{E[\min(W-t,S)|W>t]}, \tag{7}$$

and let the server choose the next customer that obtains the highest score. In other words, if there are n customers in the queue with customer i having stayed in the queue for t_i time units, this heuristic policy will next serve customer $j = \arg\max_{i=1,\ldots,n} s(t_i)$.

To further compute the denominator in Equation (7), we calculate

$$\begin{aligned} E[\min(W-t,S)|W>t] &= \int_0^\infty P\{\min(W-t,S)>x|W>t\}dx \\ &= \int_0^\infty P\{W-t>x \text{ and } S>x|W>t\}dx \\ &= \int_0^\infty P\{W-t>x|W>t\}P\{S>x|W>t\}dx \\ &= \int_0^\infty \frac{\bar{F}_W(t+x)}{\bar{F}_W(t)}\bar{F}_S(x)dx. \end{aligned} \tag{8}$$

Consequently, putting Equations (7) and (8) together gives that

$$s(t) = \frac{\bar{F}_R(t)}{\int_0^\infty \bar{F}_W(t+x)\bar{F}_S(x)dx}. \tag{9}$$

The score in Equation (9) is computed for each customer individually, based on the time he has spent in the queue. One advantage of this score is that it is easy to compute. In practice, we can compute the score $s(t)$ for all values of t beforehand, which allows easy implementation in real time. Observe that a customer's score does not depend on the number of other customers in the queue nor the customer arrival rate λ. Therefore, this heuristic cannot be optimal in general. In particular, when the traffic is relatively light, the server should take into account the possibility of being idle when the queue becomes empty, rather than always choosing the customer that yields the highest reward rate. Hence, in a light-traffic system it is possible to devise a policy that is specifically tailored for given distributions F_R, F_W, and F_S, such as the case in Example 4.1. On the other hand, in a surveillance system under heavy traffic—the case we expect to see in reality—there are many customers to choose from each time the server becomes available. Therefore, the server need not be concerned about having no customers to serve, and should focus on selecting the customer that yields the highest reward rate. Consequently, we expect our heuristic policy to be effective in a heavy-traffic system.

We next present the score function $s(t)$ in three examples summarized in Table 1. In each example, we let F_R and F_W be Erlang distribution functions, denoted by Erlang(m, β), where m is the shape parameter and β the scale parameter. In addition, we let F_S be a uniform distribution function. Each example represents a distinctive relationship between F_R and F_W.

The first example represents the case when F_R is stochastically larger than F_W. We let $F_R \sim$ Erlang(6, 1.2) and $F_W \sim$ Erlang(6, 1), so that they have the same coefficient of

Table 1: Three examples for the score function $s(t)$.

	F_R	F_W	Relationship
1	Erlang $(6, 1.2)$	Erlang $(6, 1)$	F_R stochastically larger than F_W
2	Erlang $(6, 1)$	Erlang $(6, 1.2)$	F_R stochastically smaller than F_W
3	Erlang $(6, 1)$	Erlang $(2, 3)$	F_R has smaller variance

variation. In Figure 3, we plot the score function $s(t)$ in Equation (9) when F_S is a uniform distribution with different parameters, ranging from $U(0.5, 1.5)$ to $U(7.5, 8.5)$. As seen in Figure 3, regardless of the service time distribution F_S, our heuristic policy coincides with the FCFS rule. This result is relatively easy to understand, because the longer a customer has stayed in the queue, not only is it more likely to be red, but it also tends to leave the system sooner even if it is white. Hence, the score increases in the time a customer stays in the queue.

The second example represents the case when F_R is stochastically smaller than F_W. We let $F_R \sim$ Erlang$(6, 1)$ and $F_W \sim$ Erlang$(6, 1.2)$, so that they have the same coefficient of variation. The score function $s(t)$ is plotted in Figure 4. To intuitively understand the behaviors of these curves, first note that $\bar{F}_R(t)/\bar{F}_W(t)$ (the numerator in Equation (7)) is monotonically decreasing in t, which implies that the longer a customer has stayed in the queue, the less likely it is a red customer. Second, the denominator in Equation (7) represents the expected time the server will be kept busy with the customer should it be a white customer. When the service time distribution is $U(0.5, 1.5)$, this expected time $E[\min(W - t, S)|W > t]$—where W and S are two independent random variables with respective distribution functions $F_W(\cdot)$ and $F_S(\cdot)$—remains roughly a constant for $t < 1$, because the service time distribution $F_S \sim U(0.5, 1.5)$ dictates the time the server remains busy. On the other hand, if the service time distribution $F_S \sim U(7.5, 8.5)$, then $E[\min(W - t, S)|W > t]$ decreases more significantly as t increases, because it becomes more likely the server will become free due to the premature departure of the customer under service. As a consequence, when the service time becomes stochastically larger, it becomes more desirable to serve a customer that has stayed in the queue for 6 time units or so.

The third and final example represents the case when F_R and F_W have the same mean, while F_R has a smaller variance. In particular, we let $F_R \sim$ Erlang$(6, 1)$ and $F_W \sim$ Erlang$(2, 3)$. As seen in Figure 5, $s(t)$ reaches its maximum around $t = 4$ mainly because the likelihood ratio $\bar{F}_R(t)/\bar{F}_W(t)$ exhibits a similar shape, and that a 4-time-unit-old customer is more likely to be red than other customers. A more important observation is that, in this case, the heuristic policy is significantly different from either FCFS or LCFS rules.

Finally, recall from Section 3 that if F_R is exponential with rate θ_R and F_W is exponential with rate θ_W, then the FCFS rule is optimal if $\theta_R < \theta_W$. It is straightforward to verify that in this special case our heuristic policy does yield the optimal policy.

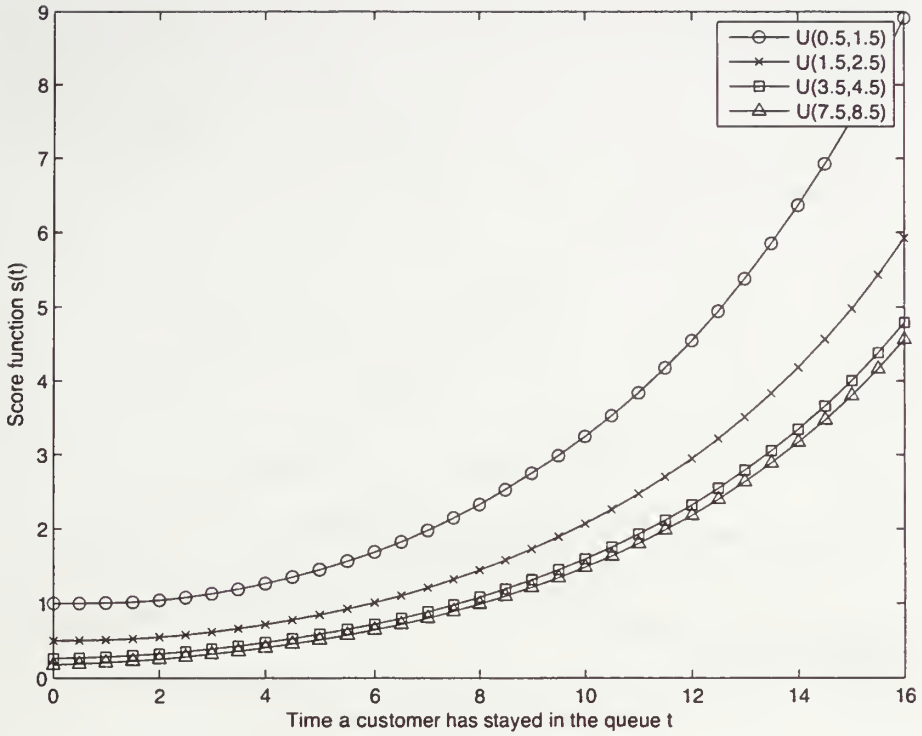

Figure 3: Score function $s(t)$; $F_R \sim \text{Erlang}(6, 1.2)$, $F_W \sim \text{Erlang}(6, 1)$, and F_S follows four different uniform distributions.

4.3 Numerical Experiments

In this subsection, we first develop a simulation algorithm to evaluate the performance of a policy, and then use the algorithm to compare different policies numerically.

To simulate the performance of a policy, first note that as $p \to 0$, our original problem is equivalent to problem B stated in the beginning of Subsection 4.2. For a given service rule, we let the white customers arrive according to a Poisson process with rate λ, and let a red customer arrive in steady state. Let $I = 1$ if that red customer enters service before leaving, and $I = 0$ otherwise. Our goal is to estimate $E[I]$, but naive estimation of $E[I]$ is very inefficient.

To overcome this issue, we generate a sample path of the queueing system where white customers arrive according to a Poisson process with rate λ for the first n arrivals, and the server processes customers according to a given service rule—FCFS, LCFS, RS, or heuristic. After generating the sample path, we turn our attention to each customer one at a time. For the jth arriving customer, define the random variable $I_j = 1$ if that customer would have received service had it been a red customer, while all other customers are white; let $I_j = 0$

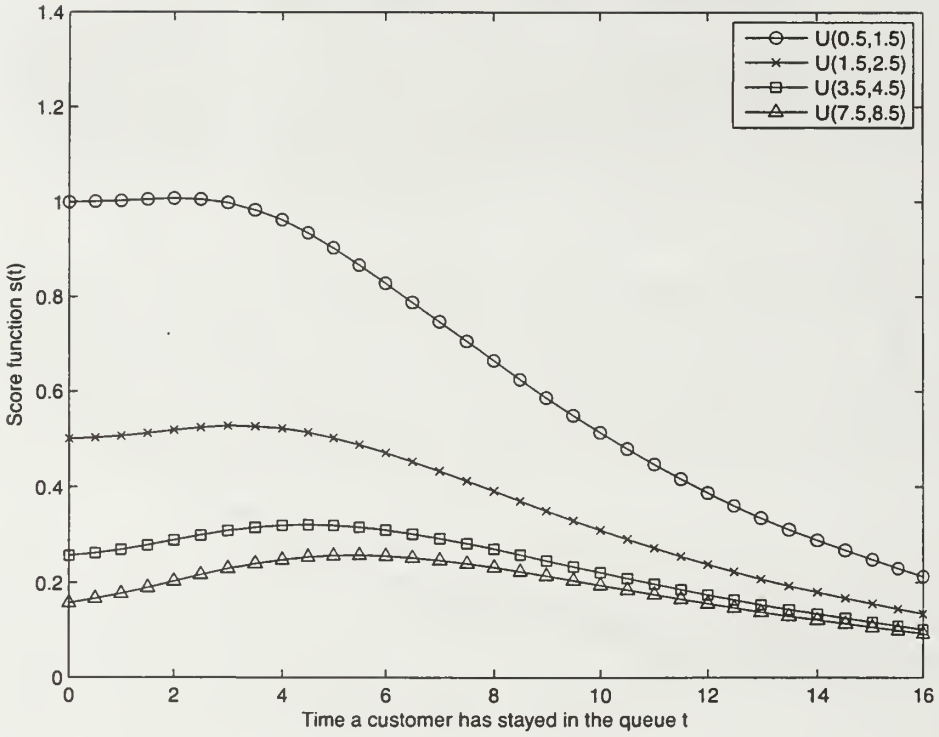

Figure 4: Score function $s(t)$; $F_R \sim$ Erlang$(6, 1)$, $F_W \sim$ Erlang$(6, 1.2)$, and F_S follows four different uniform distributions.

otherwise. Hence, our estimator is $\sum_{j=1}^{n} I_j/n$. Because white customers arrive according to a Poisson process, and because Poisson arrivals see time averages (see Wolff [18]), it follows that $(\sum_{j=1}^{n} I_j)/n$ converges almost surely to $E[I]$ as n tends to infinity.

Our simulation algorithm uses steady-state simulation to get multiple estimators in a single simulation run. There are two issues related to a steady-state simulation. First, there is initial bias because the system is not in steady state when we start the simulation with an empty queue. Second, the random variables I_j and I_{j+1} are not independent. If $I_j = 1$, it becomes more likely the queue has few customers, which in turn makes I_{j+1} more likely to take on value 1. To resolve these two issues, we allow a prolonged warm-up period before collecting data, and use batch means to estimate the standard error of our estimator; see, for example, Law and Kelton [10]. We choose the batch size so that with probability close to 1 the first customers in consecutive batches will never coexist in the system.

For a numerical experiment, we choose $F_R \sim$ Erlang$(6, 1)$, $F_W \sim$ Erlang$(2, 3)$, and $F_S \sim U(1.5, 2.5)$. Table 2 compares the probability that a red customer will receive service as $p \to 0$ for four policies when the customer arrival rate λ varies. In simulation experiments, the standard error is about 10^{-3} of the estimator. We choose the performance of the RS

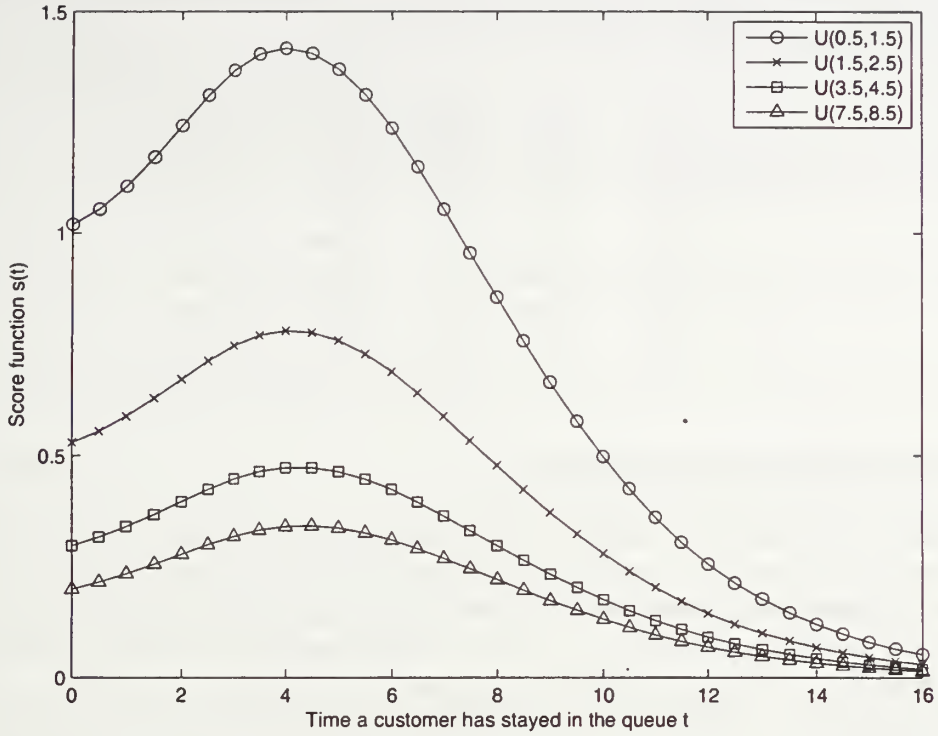

Figure 5: Score function $s(t)$; $F_R \sim \text{Erlang}(6,1)$, $F_W \sim \text{Erlang}(2,3)$, and F_S follows four different uniform distributions.

rule as the benchmark, and report the performance of the other three rules as ratios to the benchmark.

As seen in Table 2, the heuristic policy always yields the highest probability of serving a red customer. In addition, the heuristic policy's relative improvement over the RS rule gradually increases as λ increases. This observation is not surprising, as we argued in Section 4.2 that the heuristic policy is particularly suitable in a heavy-traffic system. Another interesting observation is the comparison between the FCFS rule and the LCFS rule. When $\lambda = 1$, the traffic is light, so most of the time the customers in the queue are relatively new to the queue. As seen in Figure 5, the score function $s(t)$ increases in t when t is small, so the FCFS rule performs better than the LCFS rule in a light-traffic queue. When $\lambda = 6$, the traffic is much heavier, so with the FCFS rule the server will frequently serve customers who have stayed a long time in the queue. Because the score function $s(t) < s(0)$ for $t > 8$ and with the LCFS rule most of the time the server serves a customer whose age is close to 0, the LCFS rule performs better than the FCFS rule.

Table 2: Probability of serving a red customer in different policies as $p \to 0$; $F_R \sim$ Erlang$(6,1)$, $F_W \sim$ Erlang$(2,3)$, and $F_S \sim U(1.5, 2.5)$.

		Ratio to RS Rule			
λ	RS	RS	FCFS	LCFS	Heuristic
1	0.612	1.000	1.106	0.922	1.132
2	0.311	1.000	0.963	0.900	1.230
3	0.205	1.000	0.747	0.890	1.257
4	0.153	1.000	0.586	0.888	1.270
5	0.122	1.000	0.473	0.886	1.278
6	0.101	1.000	0.390	0.885	1.281

5 Concluding Remarks

In this paper we developed a single-server queueing model with impatient customers to describe a surveillance system for detecting terrorists in a heavy-traffic arena. Two types of customers—terrorist and nonterrorist—arrive to the arena, but a customer's identity is not revealed until the server completes the service, or if the customer leaves the arena (in the case of a nonterrorist) or initiates an attack (in the case of a terrorist). The server, however, can draw inference about a customer's likely identity based on the time the customer stays in the queue. We presented a few cases in which the optimal policy can be explicitly determined, and studied a heuristic policy that performs well in a heavy-traffic system.

Because our study focused on the scheduling aspect of the screening operation, we assumed that the surveillance system has perfect sensitivity and perfect specificity. If the surveillance system would erroneously classify a terrorist as a nonterrorist (false negative) with a certain probability, then the performance of the surveillance system described in this paper would simply be discounted by that probability. If false positive errors are also possible, then the actions taken by the authorities would incur a social cost associated with the disruption of normal daily life. This cost, however, is typically much smaller than that of a successful terrorist attack.

There are a few related research directions that can follow from our study. First, the probability of classification errors can be incorporated into the model as a function of the time a target is under surveillance. The longer the surveillance system monitors a target, the more likely the classification would be correct. In this case, the service time becomes a controlled variable rather than a random parameter. Second, it is possible to extend the queueing model to allow multiple servers and more than two types of customers (e.g., terrorists and criminal fugitives). We believe that mathematical modeling along these research lines has the potential to advance the effort on counterterrorism and homeland security.

References

[1] What happened at the Lod Airport in 1972? http://www.palestinefacts.org/pf_1967to1991_lod_1972.php. Palestine Fact.

[2] Dimitris Bertsimas and Georgia Mourtzinou. Multiclass queueing systems in heavy traffic: An asymptotic approach based on distributional and conservation laws. *Operations Research*, 1997.

[3] Donald Gross and Carl M. Harris. *Fundamentals of Queueing Theory*, chapter 2.9. Wiley, 2nd edition, 1985.

[4] J. Michael Harrison. Dynamic scheduling of a multiclass queue: Discount optimality. *Operations Research*, 23(2):270–282, 1975.

[5] J. Michael Harrison and Assaf Zeevi. Dynamic scheduling of a multiclass queue in the halfin-whitt heavy traffic regime. *Operations Research*, 52(2):243–257, 2004.

[6] Sheldon H. Jacobson, Laura A. McLay, John E. Kobza, and Jon M. Bowman. Modeling and analyzing multiple station baggage screening security system performance. *Naval Research Logistics*, 52(1):30–45, 2005.

[7] Edward Kaplan and Moshe Kress. Operational effectiveness of suicide bomber detector schemes: A best-case analysis. *Proceedings of the National Academy of Sciences*, 102(29):10399–10404, 2005.

[8] Edward H. Kaplan, Christopher A. Patton, William P. FitzGerald, and Lawrence M. Wein. Detecting bioterror attacks by screening blood donors: A best-case analysis. *Emerging Infectious Diseases*, 9(8):909–914, 2003.

[9] Moshe Kress. The effect of crowd density on the expected number of casualties in a suicide attack. *Naval Research Logistics*, 52(1):22–29, 2005.

[10] Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York City, NY, 3rd edition, 2000.

[11] Laura A. McLay, Sheldon H. Jacobson, and John E. Kobza. A multilevel passenger screening problem for aviation security. *Naval Research Logistics*, 53(3):183–197, 2006.

[12] Bruce L. Miller. A queueing reward system with several customer classes. *Management Sciences*, 16(3):234–245, 1969.

[13] Tava Lennon Olsen. A practical scheduling method for multiclass production systems with setups. *Management Science*, 45(1):116–130, 1999.

[14] Erica Plambeck, Sunil Kumar, and J. Michael Harrison. A multiclass queue in heavy traffic with throughput time constraints: Asymptotically optimal dynamic controls. *Queueing Systems*, 39(1):23–54, 2001.

[15] Martin I. Reiman and Lawrence M. Wein. Dynamic scheduling of a two-class queue with setups. *Operations Research*, 46(4):532–547, 1998.

[16] Sheldon M. Ross. *Stochastic Processes.* Wiley, 2nd edition, 1996.

[17] Lawrence M. Wein and Yifan Liu. Analyzing a bioterror attack on the food supply: The case of botulinum toxin in milk. *Proceedings of the National Academy of Sciences*, 102(28):9984–9989, 2005.

[18] Ronald W. Wolff. Poisson arrivals see time averages. *Operations Research*, 30(2):223–231, 1982.

INITIAL DISTRIBUTION LIST

1. Research Office (Code 09) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 013) 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

4. Richard Mastowski (Technical Editor) 2
   Graduate School of Operational and Information Sciences (GSOIS)
   Naval Postgraduate School
   Monterey, CA 93943-5219

5. Assistant Professor Kyle Lin 1
   Operations Research Department
   Naval Postgraduate School
   Monterey, CA 93943-5219

6. Professor Moshe Kress 1
   Operations Research Department
   Naval Postgraduate School
   Monterey, CA 93943-5219

7. Assistant Professor Roberto Szechtman 1
   Operations Research Department
   Naval Postgraduate School
   Monterey, CA 93943-5219